Subject: Re: [PATCH 8/9] signal: Drop signals before sending them to init.
Posted by Oleg Nesterov on Thu, 13 Dec 2007 18:18:02 GMT

On 12/13, Eric W. Biederman wrote:
>
> Oleg Nesterov <oleg@tv-sign.ru> writes:
>
> > So, do you mean we can ignore the problems with the signals which are
> > currently blocked by /sbin/init?
>
> Yes. Further I am saying those signals will never become pending if
> we do not have a signal handler installed.

OK, if we change the semantics for /sbin/init signals we can avoid
a lot of problems,

> > I personally agree, but I'm not sure I understand this right.
> >
> >> +static int sig_init_drop(struct task_struct *tsk, int sig)
> >> +{
> >> + /* All signals for which init has a SIG_DFL handler are
> >> + * silently dropped without being sent.
> >> + */
> >> + if (!is_sig_init(tsk))
> >> + return 0;
> >> +
> >> + return (tsk->sighand->action[sig-1].sa.sa_handler == SIG_DFL);
> >> +}
> >
> > What if /sbin/init has a handler, but before this signal is delivered
> > /sbin/init does signal(SIG_DFL) ? We should modify so_sigaction() to
> > prevent this. Note again the patch above.
>
> No. We should treat signals that we process for /sbin/init completely
> normally.

... including this one. I am not arguing.

> This gives /sbin/init completely normal signal handling if the signal is
> ever enqueued. Something trivial to implement and explain.

Well, I am not sure about "explain" though. Unless I missed something
this makes the semantics a bit special.

Suppose that init does sigtimedwait() but the handler == SIG_DFL.

Oleg.

_____

Containers mailing list
Containers@lists.linux-foundation.org
https://lists.linux-foundation.org/mailman/listinfo/containers